7К

УДК 681.3

М.В. Крыжановский, М.Ю. Мальсагов
Центр оптико-нейронных технологий НИИСИ РАН, г. Москва, Россия
iont.niisi@gmail.com, Magomed.Malsagov@gmail.com

Решение задачи распределения нагрузки в двухпроцессорной системе на основе нейронных сетей*

В статье рассматривается планирование распределения потока не связанных между собой задач в многопроцессорной системе. Предлагается подход, основанный на использовании ассоциативных нейронных сетей. В случае двухпроцессорной системы его применение полностью решает поставленную задачу за $O(M)$ операций.

Введение

В последние годы широко и все более быстрыми темпами осуществляется разработка, выпуск и внедрение в автоматизированные системы обработки информации и управления, вычислительные центры и вычислительные сети разнообразных многопроцессорных и многоядерных вычислительных систем (МВС). Такие вычислительные системы отличаются повышенной гибкостью и характеризуются высокими, а в ряде случаев – рекордными показателями производительности, объемов внутренней памяти, готовности, надежности и живучести.

Программирование для многопроцессорных и многоядерных систем связано с распараллеливанием и синхронизацией вычислений и с организацией выполнения параллельных процессов. Это выдвигает целый ряд сложных задач, среди которых весьма важными являются расчет характеристик времени и количества операций, требующихся для выполнения параллельных программ, и построение расписаний (планов) выполнения параллельных программ на многопроцессорных и многоядерных вычислительных системах.

Одной из проблем, возникающих при функционировании МВС, обрабатывающих потоки задач, является балансировка нагрузки [1-6], т.е. проблема динамического распределения ресурсов системы (процессоров, ядер) по задачам потока таким образом, чтобы все процессоры (ядра) системы были загружены равномерно. Недостатки в функционировании МВС привели к созданию систем управления ресурсами (СУР), типичная схема которых приведена на рис. 1.

Она включает в себя следующие функции:
– миграция задач по узлам МВС;
– создание и функционирования множества очередей;
– планирование вычислений.

* Работа поддержана грантом РФФИ (06-01-00109).

В свою очередь, планирование вычислений определяется с помощью следующих методов:

– перечислительные, переборные методы [7];
– теоретико-графовые методы [8];
– методы математического программирования [7], или динамического программирования [9];
– методы теории очередей [10].

Однако применение перечисленных методов при динамическом планировании вычислительного процесса ограничивается их высокой вычислительной сложностью [11].

В этой связи возникает актуальная задача – создать такие методы распределения потока задач, с помощью которых можно было бы оптимизировать это распределение и обеспечить надежность функционирования в условиях деградации ресурсов системы.

Конкретный алгоритм, реализуемый планировщиком (рис. 1), в распределенных системах выполняет функцию управления ресурсами. Когда речь идет о планировании вычислений, т.е. об определении порядка выполнения процессов и работ, то нужно иметь в виду, что эта задача предполагает и назначение – распределение доступных ресурсов между процессами и работами.

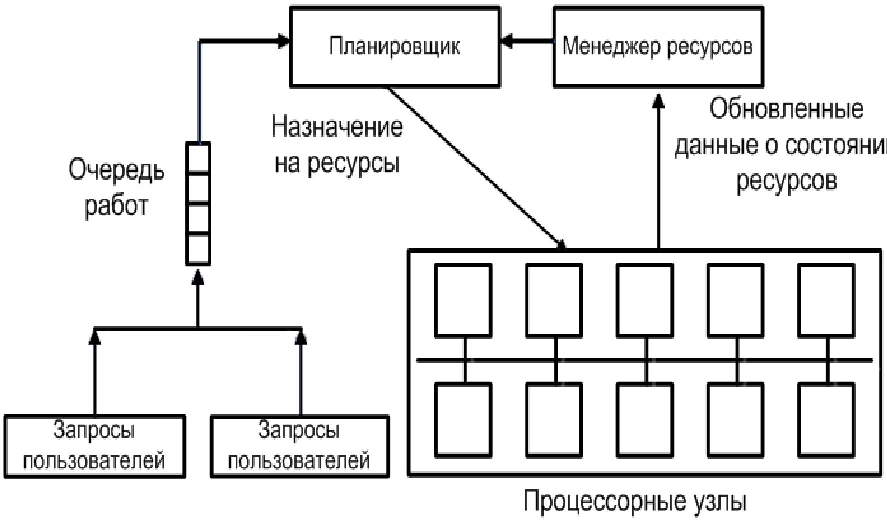


Рисунок 1 – Типичная структура системы управления ресурсами

В данной работе рассматривается метод, основанный на применении ассоциативных нейронных сетей Хопфилда [12] в алгоритме балансировки нагрузки.

Постановка задачи

Пусть имеется система, содержащая N процессоров одинаковой производительности V, на которую поступает поток из не связанных между собой M задач, каждая из которых имеет вычислительную сложность $H_i (i = 1 \ldots M)$. Тогда время выполнения процессором i-ой задачи задается выражением

$$\tau_i = H_i / V .$$

Под оптимизацией распределения задач из потока по процессорам МВС будем понимать минимизацию времени T выполнения всех задач [1], которое определяется следующим выражением

$$\mathrm{T} = \max\left\{T_1, T_2, \ldots, T_N\right\}, \quad (1)$$

где T_j – время выполнения j-ым процессором своего подмножества задач. Например, если $a, b, \ldots c$ – номера задач, подлежащих исполнению на j-ом процессоре, то это время определяется как

$$T_j = \tau_a + \tau_b + \ldots + \tau_c .$$

Необходимо найти такое разбиение множества задач потока по процессорам, чтобы значение времени T (1) было минимальным.

Переформулировка постановки задачи

Пусть G – какое-то распределение задач по процессорам МВС. Это распределение характеризуется временным параметром

$$\mathrm{T} = \mathrm{T}(G) = \max\{T_1, T_2, \ldots, T_N\}.$$

Для каждого распределения G можно вычислить дисперсию времен выполнения:

$$\sigma^2 = \frac{1}{N}\sum_{j=1}^{N}\left(T_j - \overline{T}\right)^2,$$

где

$$\overline{T} = \frac{1}{N}\sum_{j=1}^{N} T_j = \frac{1}{N}\sum_{i=1}^{M} \tau_i .$$

Тогда между параметрами σ^2и T имеется связь, о чем и свидетельствуют результаты экспериментов. Так на рис. 2 и 3 представлены результаты численного эксперимента, в котором один и тот же набор из 20 случайных задач фиксированных сложностей $H = [1, \ldots, 10]$ 10000 раз случайным образом «разбрасывался» по 4 одинаковым процессорам. Для каждого распределения задач по процессорам вычислялись время выполнения T и значение дисперсии σ^2 и эти данные наносились на график. Видно (рис. 2), что для одного и того же значения T имеется несколько значений дисперсии σ^2, однако среднее значение дисперсии $\overline{\sigma^2} = \frac{1}{K}\sum_{l=1}^{K}\sigma_l^2$ монотонно растет с увеличением T. На рис. 3 показана зависимость среднего значения $\overline{\sigma^2}$ от времени выполнения. Видно, что (усредненное по реализациям) значение дисперсии $\overline{\sigma^2}$ связано монотонной зависимостью со временем выполнения T.

Для подтверждения такой зависимости было проведено более 1000 аналогичных компьютерных экспериментов, в которых число процессоров изменялось от 2 до 10, а сложность наборов задач $M \in [6; 20]$. В этих экспериментах были получены схожие результаты, что позволяет нам перейти от минимизации величины T к минимизации дисперсии σ^2.

Отдельный интерес представляет случай распределения потока задач на двух одинаковых процессорах, поскольку:

1. В этом случае величина σ^2 является однозначной функцией времени выполнения (рис. 4) и ее уменьшение автоматически приводит к уменьшению T. В оставшейся части работы рассматривается система, состоящая из двух процессоров одинаковой производительности.

2. Этот алгоритм можно использовать в задаче оптимального управления ресурсами МВС, когда число процессоров больше 2.

3. Предложенный подход, основанный на использовании ассоциативных нейронных сетей, полностью решает поставленную задачу за $O(M)$ операций.

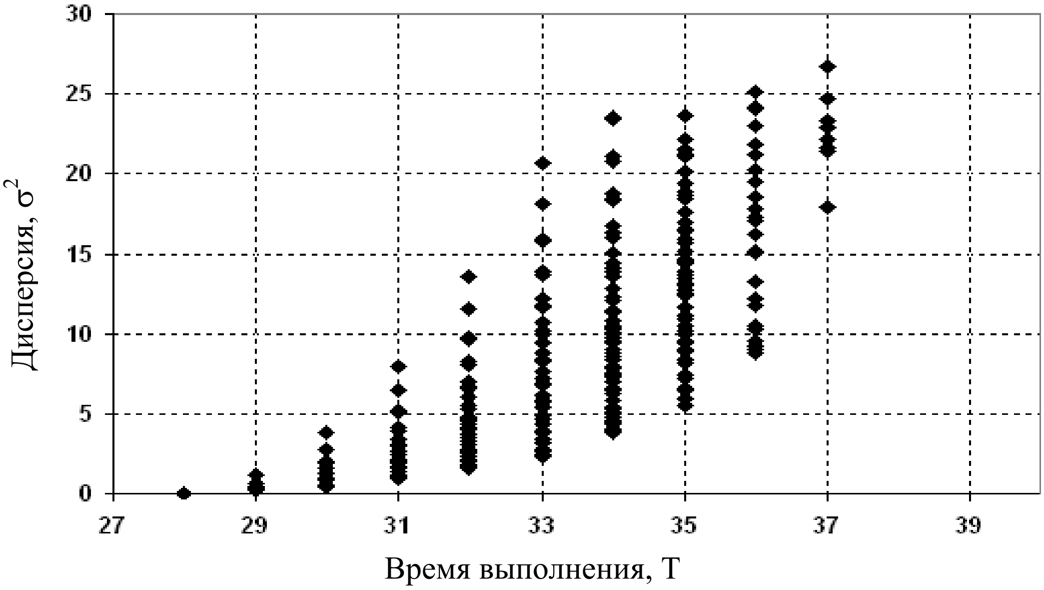


Рисунок 2 – Зависимость дисперсии от времени выполнения: 4 процессора одинаковой производительности; 20 задач, вычислительная сложность которых $H = \{4;6;8;10;3;3;8;3;5;3;8;10;9;1;1;6;7;4;7;5;\}$; число экспериментальных точек равно 10000

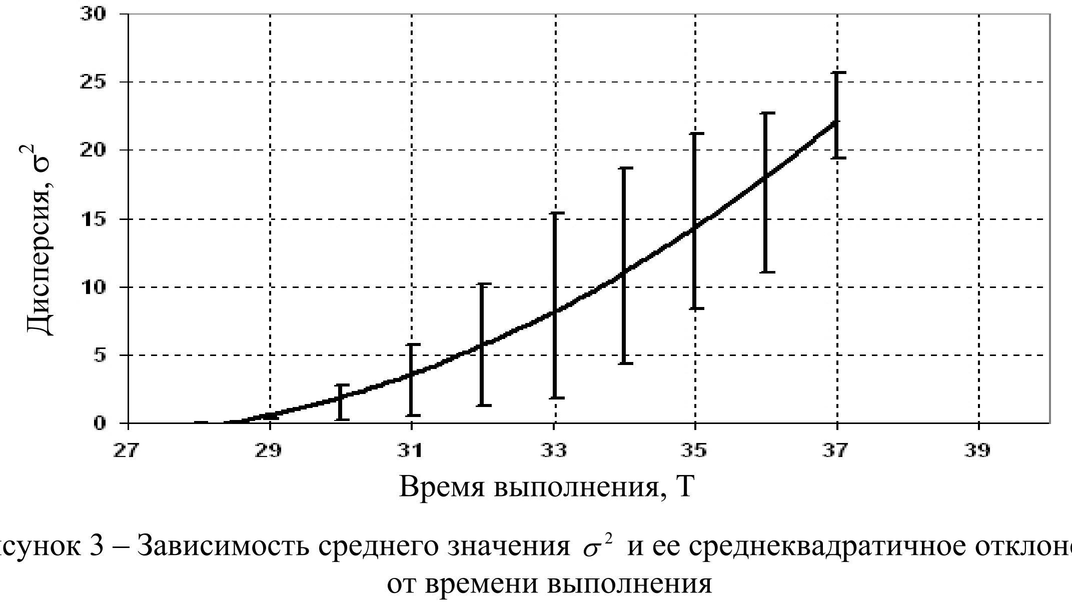


Рисунок 3 – Зависимость среднего значения σ^2 и ее среднеквадратичное отклонение от времени выполнения

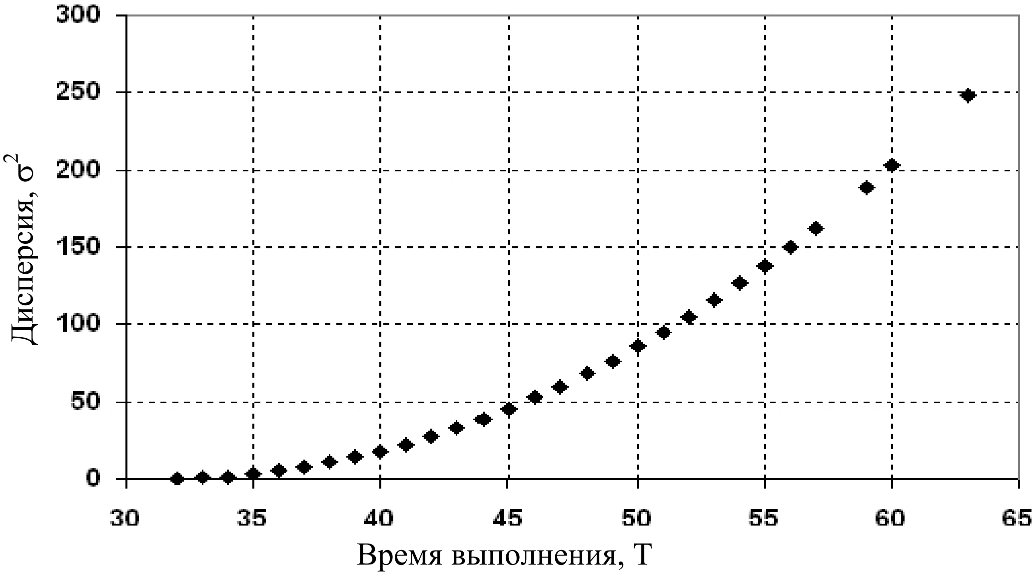


Рисунок 4 – Зависимость σ^2 от времени выполнения: 2 процессора одинаковой производительности; 10 задач, вычислительная сложность которых $H = \{7;4;7;6;7;10;8;8;3;3;\}$

Далее мы отдельно рассмотрим случаи с двумя процессорами.

Модель нейронной сети

Пусть T_1 и T_2 – времена выполнения первым и вторым процессорами своих задач соответственно. Обозначим принадлежность i-ой задачи к одному из двух одинаковых процессоров заданием бинарной переменной y_i (координаты): если $y_i = 1$, то задача подается на первый процессор, если $y_i = 0$ – на второй. В этих обозначениях выражение для дисперсии σ^2 имеет вид:

$$\sigma^2 = [T_1 - T_2]^2 = \left(\sum_{j=1}^{M} (2y_j - 1) H_j \right)^2 . \qquad (2)$$

Или, после некоторых преобразований,

$$\sigma^2 = const + \sum_{i,j=1}^{M} A_{ij} x_i x_j = \left(\sum_{i=1}^{M} H_i x_i \right)^2 - \sum_{i=1}^{M} (H_i x_i)^2 , \qquad (3)$$

где

$$A_{ij} = \begin{cases} H_i H_j , & i \neq j \\ 0, & i = j \end{cases} ,$$

$$x_i = 2y_i - 1,\ x_i = \{-1,+1\},\ \ y_i = \{0,1\} .$$

Отметим, что выражение (3) для дисперсии σ^2 совпадает с определением понятия энергии в нейросетевой модели Хопфилда [12], где в качестве $(M \times M)$-матрицы связей выступает матрица A_{ij}. Поэтому поиск минимума квадратичного функционала (3) по бинарным переменным x_i ведется с помощью стандартной нейросетевой

процедуры. Для данной структуры матрицы связей A_{ij} динамика сети Хопфилда следующая: состояние k-ого нейрона в момент времени $(t+1)$ определяется в соответствии с правилом Хопфилда:

$$x_k(t+1) = -sign(S(t) - H_k x_k(t)), \quad k = 1, \ldots, M.$$

С последующей модификацией

$$S(t+1) = S(t) - H_k x_k(t) + H_k x_k(t+1).$$

Тем самым представленная выше динамика позволяет уменьшить в M раз число операций, которое теперь составляет $O(M)$.

Компьютерное моделирование, эксперименты (дискретный случай)

Высокая скорость метода нейросетевой оптимизации обусловила его применение в задаче планирования распределения потока задач в МВС. Однако его непосредственное применение (как и всякий итерационный метод) дает результаты только при удачном выборе начального состояния, с которого система должна стартовать. Это обусловило создание метода для выбора начального состояния, с которого стартует нейросетевой алгоритм. Поэтому при компьютерном моделировании распределения потока задач в МВС использовались 3 метода, а именно:

1. **Метод выравнивания**, который решает задачу выбора начального состояния нейронной сети. Алгоритм метода заключается в следующем:

а) определяем T_0 – приближенное значение минимального времени выполнения, разделив пополам сумму всех времен исполнения задач;

б) на первый процессор подаем задачи до тех пор, пока их суммарное время выполнения не превысит значения T_0;

в) оставшиеся задачи подаются на второй процессор.

2. **Случайный поиск** – многократный поиск минимума функционала (3) с помощью нейронной сети Хопфилда, в процессе которого отбирается состояние с наименьшим значением дисперсии σ^2. Начальная точка старта (распределение задач по процессорам) выбирается случайно.

3. **Модифицированный метод балансировки нагрузки –** начальное распределение задач по процессорам определяется методом выравнивания. С полученного состояния стартует нейронная сеть.

Для получения статистически значимых результатов эксперименты проводились сериями по 1000 наборов задач при фиксированном количестве M задач в наборе и заданном диапазоне D (т.е. сложность задач H_i – это случайные целые числа в диапазоне от 1 до D). Затем серия повторялась для других значений M и D, значения которых $D = 10,\ 100;\ M = 15, 20, 60$.

Предварительно для каждого набора задач методом полного перебора отыскивался глобальный минимум функционала (3), с которым и сравнивались результаты метода выравнивания, метода случайных стартов и модифицированного метода. Результаты проведенных экспериментов приведены на рис. 5 и рис. 6.

На рис. 5 показаны графики вероятностей нахождения решений как функции расстояния от глобального экстремума при $D = 10$, $M = 20$. Для сравнения, на рис. 6 приведены аналогичные графики, при $D = 10$, $M = 15$.

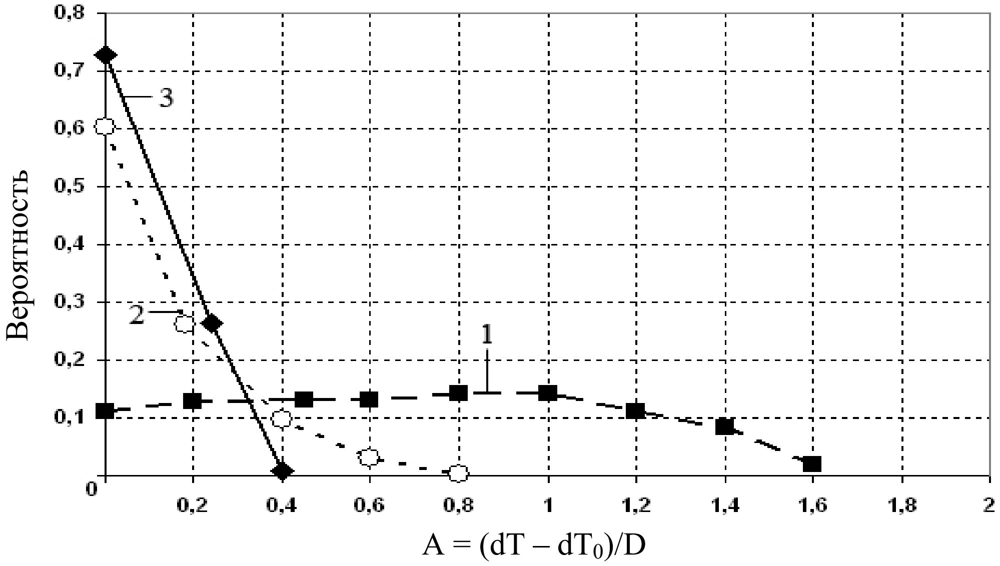


Рисунок 5 – Вероятность нахождения минимума для $M = 20$, $D = 10$ тремя методами (кривая 1 – метод выравнивания; кривая 2 – метод случайного поиска; кривая 3 – модифицированный метод балансировки нагрузки)

На всех графиках по оси абсцисс отложена переменная A, равная расстоянию до глобального минимума в единицах D, т.е. $A = (dT - dT_0)/D$, где dT_0 – разница времен выполнения для оптимального распределения, dT – разница времен выполнения для найденного распределения, A характеризует асимметрию загрузки процессоров.

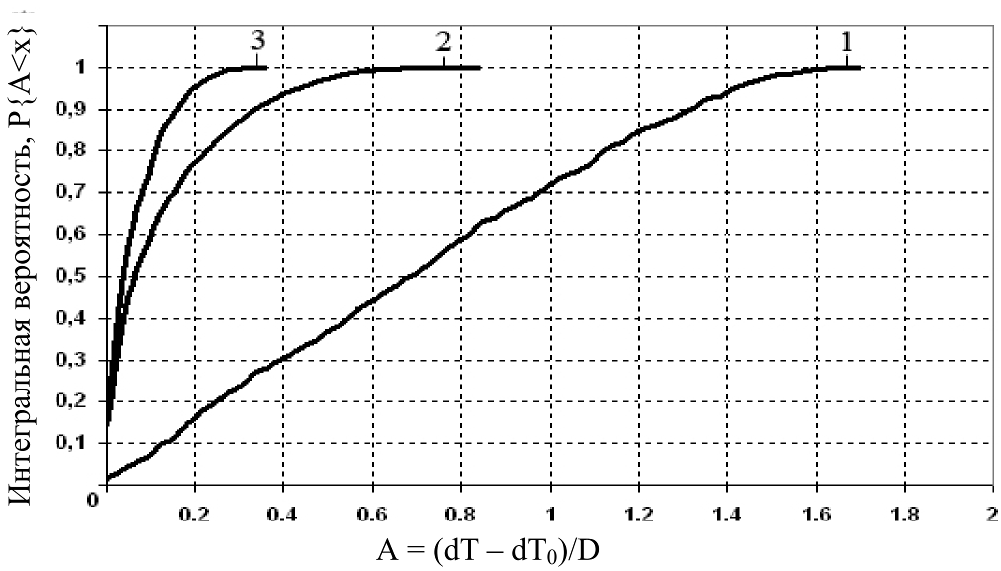


Рисунок 6 – Интегральная вероятность $P\{A < X\}$, т.е. вероятность, что асимметрия загрузки меньше X (кривая 1 – метод выравнивания; кривая 2 – случайный поиск; кривая 3 – модифицированный метод балансировки нагрузки)

Как показали эксперименты (рис. 5), непосредственное использование сети Хопфилда в методе случайных стартов дает лучшие результаты по сравнению с методом выравнивания. И в тоже время комбинация методов выравнивания и Хопфилда дает еще более сильные результаты. Этот результат нетривиальнен, так как метод Хопфилда имеет 1000 стартов на одном наборе задач, а само по себе применение «выравнивания» не эффективно. Действительно, вероятности найти оптимум на расстоянии $A \le 0{,}4$ следующие: метод выравнивания $\approx 0{,}35$; метод случайных стартов $\approx 0{,}9$; модифицированный метод ≈ 1. И для сравнения с предыдущим, вероятности на расстоянии $x \le 0{,}2$ следующие: 0,22; 0,8; 1. Серия этих экспериментов проводилась при условии, что число задач больше величины диапазона D, т.е. $M > D$.

На рис. 6 приведены результаты экспериментов (другой предельный случай), когда число задач $M = 20$ значительно меньше величины диапазона $D = 100$. По оси абсцисс отложено значение безразмерной переменной $A = \left(dT - dT_0\right)/D$ (см. выше). По оси ординат отложена вероятность того, что найденный минимум находится на расстоянии меньшем X, т.е. $P\{A < X\}$. Кривые 1, 2, 3 соответствуют разным методам поиска минимума: методу выравнивания, методу случайных стартов, модифицированному методу. И в этом предельном случае модифицированный метод (кривая 3) имеет значительное преимущество перед остальными (кривые 1 и 2). Для метода выравнивания отклонение по энергии от глобального минимума не превышает $\approx 1{,}6D$, в то время как отклонение по энергии для модифицированного метода не больше $\approx 0{,}4D$.

Сравнение результатов экспериментов для разных значений D и M показывает приблизительно одинаковые результаты. Поэтому, используя результаты экспериментов, были определены математические ожидания величины A:

$dT - dT_0 = 0{,}700D$ для метода выравнивания,

$dT - dT_0 = 0{,}126D$ для метода случайных стартов,

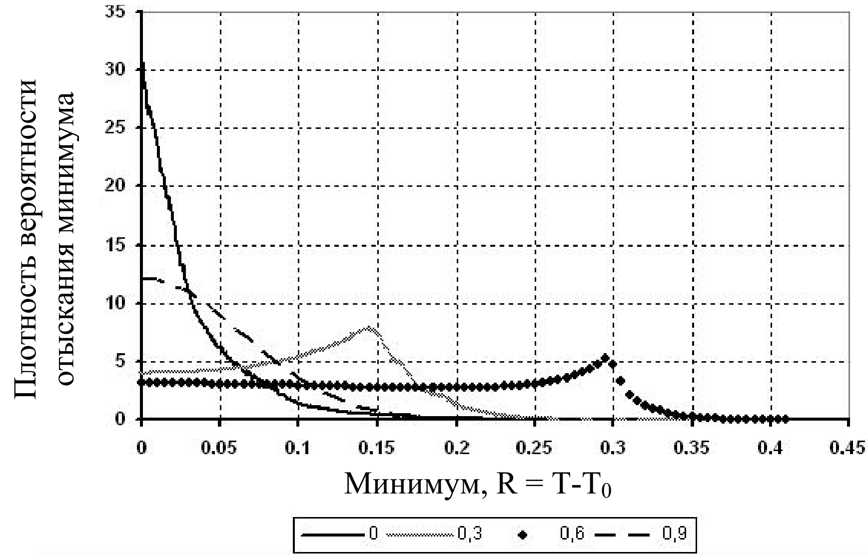
$dT - dT_0 = 0{,}065D$ для модифицированного метода.

Компьютерное моделирование, эксперименты (непрерывный случай)

Аналогичные предыдущему разделу эксперименты проводились и для непрерывного случая, т.е. в качестве сложности задач брались случайные вещественные числа в диапазоне от 0 до 1 с точностью 7 – 8 знаков после запятой. Эксперименты проводились на двух методах: **случайный поиск** и **модифицированный метод балансировки нагрузки**. Для выбора начальной точки модифицированного метода использовался **метод выравнивания**. Было сгенерировано случайным образом по 100 наборов с различным количеством задач, т.е. $M = 20; 100; 500$. Также менялась минимальная сложность задач, т.е. $H_i \in \left(D; 1\right)$, где $D = 0; 0{,}3; 0{,}6; 0{,}9$. На каждом наборе проводилось по 100000 стартов. Так как вычисление глобального минимума в данном случае невозможно, то в экспериментах мы будем следить за величиной $R = T - T_0 = T - \frac{1}{2}\sum_{i=1}^{M} H_i$, где T_0 – приближенное значение минимального времени выполнения.

На рис. 7 изображена спектральная плотность минимумов для $M = 20$ и $D = 0; 0{,}3; 0{,}6; 0{,}9$.

Модифицированный метод балансировки нагрузки



Случайный поиск

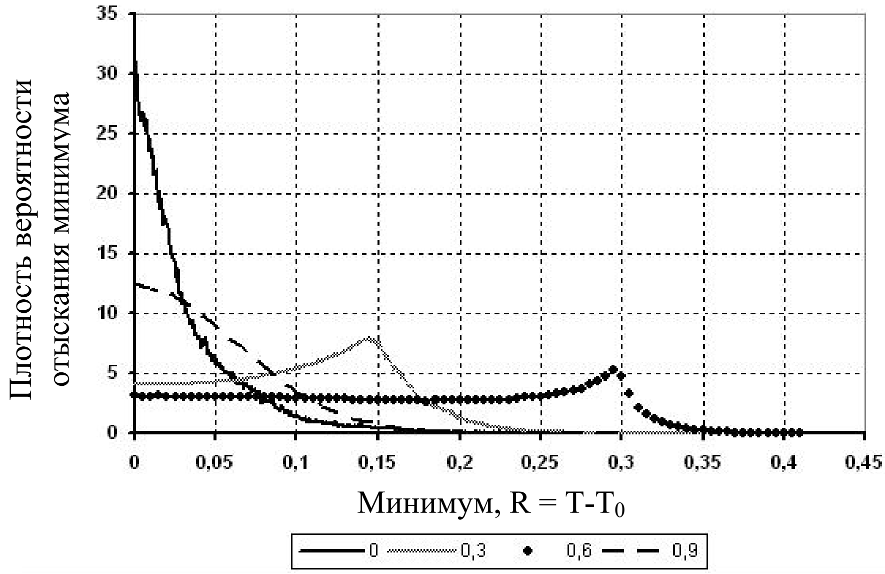


Рисунок 7 – Плотность отыскания минимума для различных значений минимальной сложности задач при $M = 20$

Средние значения величины R и значения дисперсий для описанного эксперимента приведены в табл. 1.

Таблица 1 – Средние значения R и ее дисперсия для различных минимальных сложностей задач в наборе при $M = 20$

Минимальная сложность задач, D	Модифицированный метод		Случайный поиск	
	Среднее значение	Дисперсия	Среднее значение	Дисперсия
0	0,0339	0,0013	0,0338	0,0012
0.3	0,1056	0,0031	0,1056	0,0031
0.6	0,1670	0,0095	0,1678	0,0095
0.9	0,0517	0,0015	0,0512	0,0015

Интересно, что «пики» на графиках приходятся в точки $R \approx D/2$. К сожалению, эксперименты в непрерывном случае показали, что выбор начального приближения не дает практически никакого преимущества над случайным поиском (табл. 1). Возможно, это связано с большой плотностью локальных минимумов.

Выводы

В работе были предложены алгоритмы на основе нейронных сетей, решающие задачу распределения нагрузки в двухпроцессорной системе.

Использование нейросетевой минимизации при старте со случайной точки в конфигурационном пространстве вектора x не гарантирует успеха. Для получения результата, сопоставимого с модифицированным методом балансировки нагрузки, необходимо не менее 100 стартов.

Модификация метода балансировки, когда нейросетевая минимизация сочетается с целенаправленным выбором стартовой точки, позволяет получить заметно лучшие результаты в дискретном случае, но, к сожалению, не дает практически никаких преимуществ в непрерывном случае.

Решение задачи достигается малым объемом вычислений $O(M)$, где M – размерность задачи. В большинстве методов на основе сети Хопфилда объем вычислений составляет $O(M^2)$.

Литература

1. Casavant T.L., Kuhl J.G. A taxonomy of scheduling in general-purpose distributed computing system // IEEE Trans. On Software Engineering. – 1988. – V. 14, № 2.
2. Bruno J., Coffman E.G., Sethi R. Scheduling independent tasks to reduce mean finishing time // Communications of the ACM. – 1974. – V. 17, № 7.
3. Coffman E.G., Graham R.L. Optimal scheduling for two processor systems // Acta Informatica. – 1972. – V. 1, № 3.
4. Coffman E.G., Sethi R. Algorithms minimizing mean flow-time // Acta Informatica. – 1976. – V. 6, № 1.
5. Horn W.A. Minimizing average flow time with parallel machines // Opera- Operations Research. – 1973. – V. 21, № 3.
6. Kleinrock L., Nikon A. On optimal scheduling algorithms for time-shared systems // Journal of the ACM. – 1981. – V. 28, № 3.
7. Теория расписаний и вычислительные машины / Под ред. Коффмана Э.Ф. – М.: Наука, 1984.
8. Топорков В.В. Разрешение коллизий параллельных процессов в масштабируемых вычислительных системах // Автоматика и телемеханика. – 2003. – № 5.
9. Топорков В.В. Совместное планирование и назначение процессов как метод оптимизации архитектурных решений // Автоматика и телемеханика. – 2001. – № 12.
10. Клейнрок Л. Вычислительные системы с очередями. – М.: Мир, 1979.
11. Михалевич В.С., Кукса А.И. Методы последовательной оптимизации в дискретных сетевых задачах оптимального распределения ресурсов. – М.: Наука, 1983.
12. Хайкин С. Нейронные сети: полный курс. – М.: Вильямс, 2005.

М.В. Крижановський, М.Ю. Мальсагов

Вирішення завдання розподілу навантаження у двопроцесорній системі на основі нейронних мереж

У статті розглядається планування розподілу потоку не пов'язаних між собою завдань у багатопроцесорній системі. Пропонується підхід, що ґрунтується на використанні асоціативних нейронних мереж. У випадку двопроцесорної системи його застосування цілком вирішує поставлене завдання за $O(M)$ операцій.

M.V. Kryzhanovsky, M.U. Mal'sagov

Problem Solution of Distribution Stream of Tasks in System with Two Processors Based on Neural Networks

In the paper a scheduling distribution stream of disappeared tasks in multiprocessor system problem is considered. The approach which based on the associative neural network is proposed. In the system with two processors, this method solves the problem in $O(M)$ operations.

Статья поступила в редакцию 01.07.2008.